#19
S. Bond
8/6/03

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of: Boris KLOTS, et al.	Confirmation No.: 2232
Serial No.: 09/298,603	Examiner: Thong H. Vu
Filing Date: April 23, 1999	Art Unit: 2142
For: COMMUNICATION ARCHITECTURE FOR DISTRIBUTED COMPUTING ENVIRONMENT	Customer No.: 29989

Mail Stop Appeal Brief – Patents
Commissioner for Patents
P. O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

Sir:

This Appeal Brief is submitted in support of the Notice of Appeal filed June 17, 2003.

I. REAL PARTY IN INTEREST

Oracle Corporation is the real party in interest.

II. RELATED APPEALS AND INTERFERENCES

Appellants are unaware of any related appeals or interferences.

III. STATUS OF CLAIMS

Claims 1-4, 6-9, 16-23, 25-28 and 30-32 are pending in this application and have been finally rejected. Specifically, Claims 1-4, 6-9, 16-23, 25-28 and 30-32 were finally rejected under 35 U.S.C. § 103(a) as being unpatentable over *Zuili et al.,* U.S. Patent No. 6,145,084 (hereinafter "*Zuili*"). Claims 1-4, 6-9, 16-23, 25-28 and 30-32 were also finally rejected under 35 U.S.C. § 102(e) as being anticipated by *Chessell*, U.S. Patent No. 6,324,589.

IV. STATUS OF AMENDMENTS

The claims were not amended after the Final Office Action.

V. SUMMARY OF THE INVENTION

In a distributed computing environment, work can be performed on multiple computing entities, often referred to as "nodes." One example of a node is a computer system on which one or more processes or threads may execute. In the context of distributed computing, the term "work" refers to any processing of data. An important concern in distributed computing is how to provide communications between processes executing on different nodes to ensure that work is completed in an orderly and efficient manner. Two conventional approaches for providing communications between processes on different nodes include the direct communication architecture approach (see FIG. 1 and text at Page 1, line 18 through Page 3, line 5 of specification) and the managed communication architecture approach (see FIG. 2 and text at Page 3, line7, through Page 4, line 12 of the specification).

According to the direct communication architecture approach, client processes communicate directly with server processes on other nodes to request that work be performed and to obtain work results. FIG. 1 of the specification depicts a distributed computing system 100 that uses the direct communication architecture approach. In distributed computing system 100, nodes N1 and N2 are client nodes while nodes N3-N5 are server nodes. When a particular client process on client node N1 requires work to be performed on server node N3, the particular client process sends a request directly to a server process on node N3 over connection 102. The request may be in any form, for example a remote procedure call (RPC). The process on server node N3 performs the requested work and provides the results directly to the particular client process. Work is similarly performed on server nodes N3-N5 on behalf of client processes on client node N2.

A disadvantage of the direct communication architecture approach is that client processes must know which server process can perform the required work so that the client processes know where to send their requests for work. Thus, when the services offered by any of the server processes change, the client nodes must be made aware of the change. Similarly, when a new service is provided, the client processes must be made aware of the server process that provides the new service. For distributed computing systems with a large number of client processes and server nodes, a significant amount of system resources can be required to keep all the client processes updated.

According to the managed communication architecture approach, client processes send work requests to an intermediary management process referred to herein as a "director," instead of directly to the server process. The director forwards the work

requests to the appropriate server process that can perform the required work. After performing the work, the server processes send the work results to the director, which forwards the work results to the client process that made the original work request.

FIG. 2 of the specification depicts a distributed computing system 200 that uses a managed communication architecture approach. Client processes on client nodes N1 and N2 submit work requests to director 202 via connections 204 and 206. Director 202 determines which of the server nodes N3-N5 can perform the requested work and forwards the work requests to the appropriate server process on server node N3-N5 via connections 208, 210 and 212, respectively. After the appropriate server process has performed the requested work, the server process provides the work results to director 202. Director 202 then forwards the work results to the client process that requested the work.

According to the managed communication architecture approach, director 202 keeps track of the services provided by server nodes N3-N5. As a result, when a change in services offered by server nodes N3-N5 occurs, only director 202 has to be informed of the change, and not client processes on client nodes N1 and N2. This reduces the amount of system resources required to keep track of changes in services offered by server nodes N3-N5, particularly when there are a large number of client and server nodes. However, the managed communication architecture approach requires more system resources than the direct communication architecture approach to manage the additional messaging traffic associated with director 202. For example, for a single work request, the direct communication architecture approach requires two messages, one message from a client process to a server process and one message from the server process back to the client

process. To process the same work request, the managed communication architecture approach requires four messages, one from the client process to the director, one from the director to the server process, one from the server process back to the director and one from the director back to the client process.

The inventions recited in Claims 1-4, 6-9, 16-23, 25-28 and 30-32 of the present application address the problem of how to eliminate the need for client processes to be aware of the services provided by the server processes and nodes, as is required by the direct communication architecture approach. The inventions recited in Claims 1-4, 6-9, 16-23, 25-28 and 30-32 of the present application also require relatively fewer system resources than the managed communication architecture approach.

According to the inventions recited in Claims 1-4, 6-9, 16-23, 25-28 and 30-32 of the present application, client processes on client nodes send work requests to a director. The director examines the work requests to determine resources and/or services that are required to perform the requested work. The director then forwards the work requests to server processes on server nodes that can perform the requested work. The work requests provided to the server processes specify that work results are to be provided directly from the server processes to the client processes that made the work requests. Consequently, server processes provide their work results directly to the client processes that requested the work, thereby bypassing the director, even though the server processes received the work request from the director. This approach eliminates the need for client processes to be aware of the services provided by the server processes since the director determines the resources and services required to process requests. This approach also requires

relatively fewer system resources than the managed communication architecture approach since fewer messages are required.

VI. ISSUES

Issue #1: Whether Claims 1-4, 6-9, 16-23, 25-28 and 30-32 are patentable under 35 U.S.C. § 103(a) over *Zuili*.

Issue #2: Whether Claims 1-4, 6-9, 16-23, 25-28 and 30-32 are patentable under 35 U.S.C. § 102(e) over *Chessell*.

VII. GROUPING OF CLAIMS

The claims should not be regarded as all standing together since the claims recite respective limitations that render each claim separately patentable. For this appeal, the following groups are recognized:

GROUP 1: Claims 1-4, 6-9, 16-23 and 25-28.

GROUP 2: Claims 30-32.

VIII. ARGUMENT

A. Claims 1-4, 6-9, 16-23 and 25-28 (GROUP 1) are patentable under 35 U.S.C. § 103(a) over *Zuili* because Claims 1-4, 6-9, 16-23 and 25-28 include one or more limitations that are not taught or suggested by *Zuili*

CLAIM 1

Claim 1 recites a method for processing data on a distributed computing system that includes a plurality of nodes that requires the steps of:

"maintaining mapping data that specifies work that can be performed by each of the plurality of nodes;
in response to receiving a first work request to perform first work from a first process on a first node from the plurality of nodes, determining based upon the first work and the mapping data, that the first work is to be performed on a second node from the plurality of nodes;
providing the first work request to a second process on the second node, wherein the first work request specifies that the first process is to directly receive results of the first work;
determining based upon the first work and the mapping data, that the first work is also to be performed on a third node from the plurality of nodes, and
providing a second work request to a third process on the third node, wherein the second work request specifies that results of the first work performed on the third node are to be provided directly to the first process."

It is respectfully submitted that Claim 1 includes at least several limitations that are not taught or suggested by *Zuili*. As one example, it is respectfully submitted that *Zuili* does not teach or suggest "maintaining mapping data that specifies work that can be performed by each of the plurality of nodes," as recited in Claim 1. The Final Office Action asserted that the protocol conversion and language translation data maintained by verifying server 12, as described in *Zuili* at Col. 4, line 43 through Col. 5, line 2, is the "mapping data" recited in Claim 1.

As set forth in the Reply to the Final Office Action filed on April 18, 2003, the text at this portion of *Zuili* describes that verifying server 12 maintains "information as to the capability level of the sending and/or receiving devices, along with application programs necessary to translate the signals that may be sent by that device to an alternate form which may be required for the receiving device." For example, the data maintained by verifying server 12 may specify whether devices A and B are permitted to communicate with each other. This data may include "information contained in a database which contains a set of permissible recipient devices for device A" (Col. 5, lines 23-26). The data maintained by verifying server 12 may also specify adaptation, protocol

conversion or language translations that are required for devices to communicate with each other. For example, the data maintained by verifying server 12 may specify that a particular language translation is required for device A to communicate with device B. Verifying server 12 uses the data to authorize requests for communications between devices.

An example scenario is described in *Zuili* as follows. When device A wishes to communicate with device B, device A sends a Request to Send (RTS) command to verifying server 12. When verifying server 12 determines that device A is authorized to communicate with device B, verifying server 12 generates and provides a send authorization signal to device A. Device B generates and sends a Request to Receive Authorization signal to verifying server 12. Verifying server 12 compares the Request to Receive Authorization signal against a database to determine whether device B is authorized to receive the communication signal from device A. If so, then verifying server 12 generates and sends a receive authorization signal to device B. Device B then begins receiving communications from device A.

Zuili does not teach or suggest, however, that the data maintained by verifying server 12 indicates the work that can be performed by devices A and B. It is therefore respectfully submitted that the data maintained by the verifying server 12 of *Zuili* cannot be the "mapping data" recited in Claim 1, since this data does not specify the "work that can be performed by each of the plurality of nodes," as is required by Claim 1. This argument was set forth in the Reply to the Final Office Action filed on April 18, 2003. The Examiner provided a response to these arguments on Page 2 of the Advisory Action mailed on May 5, 2003, as follows:

> Applicant argues the prior art did not teach mapping data perform by every nodes. Examiner notes the prior art taught the first node (client) sends request to verification server which compared (mapped) to the stored database [*Zuili* Col. 3, lines 53-63]. It is clearly the destination node [device B 16, *Zuili* Fig. 1] received the mapping data, processed and responded to the requestor [device A 14, *Zuili* Fig. 1]. *Zuili* taught the verifying server may be mainframe, server or local computer [*Zuili* Col. 4, lines 56-Col. 5, lines 2]. Thus, every nodes performed mapping data or compare requests to its database.

Regarding the statement in the Advisory Action, "[i]t is clearly the destination node [device B 16, *Zuili* Fig. 1] received the mapping data, processed and responded to the requestor [device A 14, *Zuili* Fig. 1]," *Zuili* does not teach or suggest that device B receives the mapping data as defined by the Examiner, i.e., that device B receives the data maintained by verifying server 12. Furthermore, Claim 1 is distinguished over *Zuili* not because the mapping data as defined by the Examiner, i.e., the data maintained by verifying server 12, is not provided to the third node, i.e., device B, but because the mapping data defined by the Examiner does not satisfy the requirements of Claim 1, i.e., that the mapping data does not specify the "work that can be performed by each of the plurality of nodes." Regarding the statement in the Advisory Action, "*Zuili* taught the verifying server may be mainframe, server or local computer [*Zuili* Col. 4, lines 56-Col. 5, lines 2]. Thus, every nodes performed mapping data or compare requests to its database," it is unclear what the type of computing platform used for the verifying server has to do with whether the mapping data, as defined by the Examiner, teaches or suggests the mapping data limitations required by Claim 1. In view of the foregoing, it is respectfully submitted that the Examiner has not met his burden in showing that the step

of "maintaining mapping data that specifies work that can be performed by each of the plurality of nodes" recited in Claim 1 is taught or suggested by *Zuili.*

As another example, it is respectfully submitted that *Zuili* does not teach or suggest performing the first work on both the second and third nodes, as is required by Claim 1. The Final Office Action asserted that the "first node" recited in Claim 1 is device A of FIG. 1, the "second node" recited in Claim 1 is verifying server 12 and that the "third node" recited in Claim 1 is device B. The Final Office Action also asserted that the "first work request" recited in Claim 1 is the request-to-send (RTS) that device A sends to verifying server 12 when device A desires to communicate with device B. The Office Action further asserted that the "first work" recited in Claim 1 is the verification process performed by verifying server 12.

Given these assertions, the requirements in Claim 1 that the first work be performed on both the second and third nodes is not in any way taught or suggested by *Zuili* since device B does not perform the verification processes performed by verifying server 12. Thus, the "first work" of *Zuili*, i.e., the verification process performed by verifying server 12, is performed only on the "second node," i.e., verifying server 12, and not on the "third node," i.e., device B. It is therefore further submitted that the Examiner has not met his burden of showing that *Zuili* teaches or suggests performing the first work on both the second and third nodes, as is required by Claim 1.

Claim 1 also requires that results of the first work performed on the second and third nodes be provided directly to the first process on the first node, per the two "providing" steps. In *Zuili*, verifying server 12 provides the send authorization signal to device A. Even if device B were to perform the verification function of verifying server

12, *Zuili* does not teach or suggest that device B provides any work results to device A. It is therefore respectfully submitted that the additional requirements in Claim 1 that results of the first work performed on the second and third nodes be provided directly to the first process on the first node is also not taught or suggested by *Zuili*.

In view of the foregoing, it is respectfully submitted that Claim 1 contains several limitations that are not taught or suggested by *Zuili* and Claim 1 is therefore patentable over *Zuili*.

CLAIMS 2-4 AND 6-9

Claims 2-4 and 6-9 depend from Claim 1 and include all of the limitations of Claim 1. It is therefore respectfully submitted that Claims 2-4 and 6-9 are patentable over *Zuili* for at least the reasons set forth herein with respect to Claim 1.

CLAIM 16

Claim 16 recites limitations similar to Claim 1, except in the context of a distributed computing system for performing work. It is therefore respectfully submitted that Claim 16 is patentable over *Zuili* for at least the reasons set forth herein with respect to Claim 1.

CLAIMS 17-19

Claims 17-19 depend from Claim 16 and include all of the limitations of Claim 16. It is therefore respectfully submitted that Claims 17-19 are patentable over *Zuili* for at least the reasons set forth herein with respect to Claim 16.

CLAIMS 20-23 AND 25-28

Claims 20-23 and 25-28 recite limitations similar to Claims 1-4 and 6-9, except in the context of computer-readable media. It is therefore respectfully submitted that Claims 20-23 and 25-28 are patentable over *Zuili* for at least the reasons set forth herein with respect to Claims 1-4 and 6-9.

Based on the foregoing, it is respectfully submitted that Claims 1-4, 6-9, 16-23 and 25-28 are patentable under 35 U.S.C. § 103(a) over *Zuili.*

B. Claims 30-32 (GROUP 2) are patentable under 35 U.S.C. § 103(a) over *Zuili* because Claims 30-32 include one or more limitations that are not taught or suggested by *Zuili*

It is respectfully submitted that Claims 30-32 are separately patentable from Claims 1-4, 6-9, 16-23 and 25-28 because Claims 30-32 contain limitations that are both not required by Claims 1-4, 6-9, 16-23 and 25-28 and are not in any way taught or suggested by *Zuili.*

CLAIM 30

Claim 30 recites a method for processing data on a distributed computing system that includes a plurality of nodes that requires the steps of:

> "maintaining mapping data that specifies work that can be performed by each of the plurality of nodes; and
> in response to receiving a first work request to perform first work from a first process on a first node from the plurality of nodes,
> determining based upon the first work and the mapping data, that the first work is to be performed on a second node from the plurality of nodes,

generating an updated first work request that specifies that the first process is to directly receive results of performing the first work, and providing the updated first work request to a second process on the second node."

It is respectfully submitted that Claim 30 includes at least several limitations that are not in any way taught or suggested by *Zuili*. For example, Claim 30 requires the step of "maintaining mapping data that specifies work that can be performed by each of the plurality of nodes." This identical limitation is required by Claim 1 and, as described herein with respect to Claim 1, is not taught or suggested by *Zuili*.

Claim 30 also requires "in response to receiving a first work request to perform first work from a first process on a first node from the plurality of nodes, determining based upon the first work and the mapping data, that the first work is to be performed on a second node from the plurality of nodes, generating an updated first work request that specifies that the first process is to directly receive results of performing the first work, and providing the updated first work request to a second process on the second node."

Given the assertions in the Final Office Action as to the elements in *Zuili* that constitute the "first node," "second node" and "first work," verifying server 12 would have to, in response to receiving an RTS from device A, determine, based upon the protocol conversion and language translation functions, that verifying server 12 is to perform a verification function and then generate and provide to itself an updated request to perform the verification function. *Zuili* describes that in response to receiving an RTS from device A, verifying server 12 generates and provides an authorization signal back to device A. There is no mention or suggestion in *Zuili* that verifying server 12 generates an updated RTS and provides the updated RTS to itself. There is no discernable support in either the Final Office Action or the Advisory Action for the assertion that *Zuili* teaches

or suggests the limitations of "generating an updated first work request that specifies that the first process is to directly receive results of performing the first work, and providing the updated first work request to a second process on the second node," that are required by Claim 30.

Based on the foregoing, it is respectfully submitted that Claim 30 includes at least several limitations that are not in any way taught or suggested by *Zuili* and that the Examiner has not met his burden in showing that all of the limitations required by Claim 30 are taught or suggested by *Zuili*. It is therefore respectfully submitted that Claim 30 is patentable over *Zuili*.

Claims 31 and 32 recite limitations similar to Claim 30, except in the context of an apparatus and computer-readable medium, respectively. It is therefore respectfully submitted that Claims 30-32 are patentable under 35 U.S.C. § 103(a) over *Zuili* for at least the reasons set forth herein with respect to Claim 30.

C. Claims 1-4, 6-9, 16-23 and 25-28 (GROUP 1) are patentable under 35 U.S.C. § 102(e) over Chessell because Claims 1-4, 6-9, 16-23 and 25-28 include one or more limitations that are not taught or suggested by Chessell

CLAIM 1

Claim 1 recites a method for processing data on a distributed computing system that includes a plurality of nodes that requires the steps of:

"maintaining mapping data that specifies work that can be performed by each of the plurality of nodes;

in response to receiving a first work request to perform first work from a first process on a first node from the plurality of nodes, determining based upon the first work and the mapping data, that the first work is to be performed on a second node from the plurality of nodes;

providing the first work request to a second process on the second node, wherein the first work request specifies that the first process is to directly receive results of the first work;
determining based upon the first work and the mapping data, that the first work is also to be performed on a third node from the plurality of nodes, and
providing a second work request to a third process on the third node, wherein the second work request specifies that results of the first work performed on the third node are to be provided directly to the first process."

It is respectfully submitted that Claim 1 includes several limitations that are not in any way taught or suggested by *Chessell.* For example, the step of "maintaining mapping data that specifies work that can be performed by each of the plurality of nodes" is not in any way taught or suggested by *Chessell.* As set forth in the Reply to the Final Office Action filed on April 18, 2003, assuming that the data update logic process 33 of *Chessell* is a node on which work can be performed, no mapping data is maintained that specifies work that can be performed by data update logic process 33. There is no teaching or suggestion in Chessell whatsoever of "maintaining mapping data that specifies work that can be performed by each of the plurality of nodes."

In the Advisory Action mailed on May 5, 2003, the Examiner stated "[a]pplicant argues the prior art did not teach mapping data performed by every nodes." Applicant has never made this assertion and does not do so now. Furthermore, the claims do not require "mapping data performed by every nodes." The claims do require "maintaining mapping data that specifies work that can be performed by each of the plurality of nodes," and this limitation is not taught or suggested by *Chessell* for the reasons stated above.

The Advisory Action also asserted "Examiner notes the prior art taught each server object involved in the transaction can be told to commit the changes (or mapping data) made locally to the local database associated with server object [*Chessell* Col. 3,

lines 34-50]. It is clearly that each server performed mapping data or commit the data changes to the local database." The text at Col. 3, lines 34-50 of *Chessell* describes one of the fundamental tenets of an atomic transaction, namely, that either all of the changes are made or none of the changes are made. According to this portion of *Chessell*, this is accomplished by each server object involved in a distributed transaction registering with a coordinator object. The coordinator object coordinates the processing of the transaction to ensure that all server objects involved in the transaction either commit their changes or to undo their changes. Based on this section of *Chessell*, the coordinator object knows the identity of server objects involved with a transaction by virtue of the registration process. Presumably the registration process also provides sufficient information so that the coordinator object knows how to communicate with each of the participating server objects. This does not necessarily mean, however, that the coordinator object knows the work that can be performed by the participating server objects and in fact, there is no reason for the coordinator object to know this. Rather, the coordinator object need only know which server objects are participating in a particular transaction and how to communicate with those server objects, so that the coordinator object can instruct the participating server objects to either commit their changes or to undo their changes. For this reason, absent some teaching or suggestion in *Chessell*, it cannot be presumed that the coordinator object maintains mapping data as recited in Claim 1. It is therefore respectfully submitted that the mapping data limitation recited in Claim 1 is not taught or suggested by *Chessell* and the Examiner has not met his burden in showing that *Chessell* teaches this limitation.

It is also respectfully submitted that the limitation of "in response to receiving a first work request to perform first work from a first process on a first node from the plurality of nodes, determining based upon the first work and the mapping data, that the first work is to be performed on a second node from the plurality of nodes" is also not in any way taught or suggested by *Chessell*. In *Chessell*, no determination is made that a particular node from a plurality of nodes is to perform the requested work based upon the work request and mapping data. In *Chessell*, the work is always performed by the data update logic process 33.

Claim 1 also requires that "the first work request specifies that the first process is to directly receive results of the first work." In *Chessell*, user interface process 31 does not receive results of work performed by data update logic process 33. The only data received by user interface process 31 from data update logic process 33 is a registration request so that data update logic process 33 can participate in two-phase commit transactions.

Claim 1 also requires that a determination is made "that the first work is also to be performed on a third node from the plurality of nodes." Thus, the first work is performed on both the second and third nodes. *Chessell* does not in any way teach or suggest this limitation. In *Chessell*, work is only performed on data update logic process 33. The Examiner has not, to date, proffered any basis for how *Chessell* teaches this limitation.

Based on the foregoing, it is respectfully submitted that Claim 1 includes at least several limitations that are not in any way taught or suggested by *Chessell* and is therefore patentable over *Chessell*.

CLAIMS 2-4 AND 6-9

Claims 2-4 and 6-9 depend from Claim 1 and include all of the limitations of Claim 1. It is therefore respectfully submitted that Claims 2-4 and 6-9 are not anticipated by *Chessell* for at least the reasons set forth herein with respect to Claim 1. Furthermore, it is respectfully submitted that Claims 2-4 and 6-9 recite additional limitations that independently render them patentable over *Chessell.*

CLAIMS 16-19

Claims 16-19 include limitations similar to Claims 1-4 and 6-9, except in the context of a distributed computing system. It is therefore respectfully submitted that Claims 16-19 are not anticipated by *Chessell* for at least the reasons set forth herein with respect to Claims 1-4 and 6-9.

CLAIMS 20-23 AND 25-28

Claims 20-23 and 25-28 include limitations similar to Claims 1-4 and 6-9, except in the context of a computer-readable medium. It is therefore respectfully submitted that Claims 20-23 and 25-28 are not anticipated by *Chessell* for at least the reasons set forth herein with respect to Claims 1-4 and 6-9.

D. Claims 30-32 (GROUP 2) are patentable under 35 U.S.C. § 102(e) over *Chessell* because Claims 30-32 include one or more limitations that are not taught or suggested by *Chessell*

It is respectfully submitted that Claims 30-32 are separately patentable from Claims 1-4, 6-9, 16-23 and 25-28 because Claims 30-32 contain limitations that are both

not required by Claims 1-4, 6-9, 16-23 and 25-28 and are not in any way taught or suggested by *Chessell.*

CLAIMS 30-32

It is respectfully submitted that Claim 30 includes at least several limitations that are not in any way taught or suggested by *Chessell.* For example, Claim 30 requires the step of "maintaining mapping data that specifies work that can be performed by each of the plurality of nodes." This step is also required by Claim 1, and as described herein, is not taught or suggested by *Chessell.*

Claim 30 also requires the steps of "in response to receiving a first work request to perform first work from a first process on a first node from the plurality of nodes, determining based upon the first work and the mapping data, that the first work is to be performed on a second node from the plurality of nodes," which are also not in any way taught or suggested by *Chessell.* In *Chessell*, assuming that the data update logic process 33 is a node on which work can be performed, no mapping data is maintained that specifies work that can be performed by data update logic process 33. Furthermore, in *Chessell*, no determination is made that a particular node from a plurality of nodes is to perform the requested work based upon the work request and mapping data. In *Chessell*, the work is always performed by the data update logic process 33.

As another example, Claim 30 also requires "generating an updated first work request that specifies that the first process is to directly receive results of performing the first work" and "providing the updated first work request to a second process on the second node." These limitations are not in any way taught or suggested by *Chessell.* In *Chessell*, the transaction context is provided by business logic process 32 to data update

logic process 33 without modification. There is no discernable support in either the Final Office Action or the Advisory Action for the assertion that *Chessell* teaches or suggests the limitations of "generating an updated first work request that specifies that the first process is to directly receive results of performing the first work, and providing the updated first work request to a second process on the second node," that are required by Claim 30.

Based on the foregoing, it is respectfully submitted that Claim 30 includes at least several limitations that are not in any way taught or suggested by *Chessell* and that Claim 30 is patentable over *Chessell*. Furthermore, the Examiner has not met his burden in showing that all of the limitations required by Claim 30 are taught or suggested by *Chessell*.

Claims 31 and 32 recite limitations similar to Claim 30, except in the context of an apparatus and computer-readable medium, respectively. It is therefore respectfully submitted that Claims 30-32 are patentable under 35 U.S.C. § 102(e) over *Chessell* for at least the reasons set forth herein with respect to Claim 30.

///

///

///

IX. CONCLUSION AND PRAYER FOR RELIEF

Based on the foregoing, it is respectfully submitted that the rejections of Claims 1-4, 6-9, 16-23, 25-28 and 30-32 under 35 U.S.C. § 103(a) and 35 U.S.C. § 102(e) lack the requisite factual and legal bases. Appellants therefore respectfully request that the Honorable Board reverse the rejection of Claims 1-4, 6-9, 16-23, 25-28 and 30-32 under 35 U.S.C. § 103(a) over *Zuili* and the rejection of Claims 1-4, 6-9, 16-23, 25-28 and 30-32 under 35 U.S.C. § 102(e) over *Chessell.*

Date: July 31, 2003

Respectfully submitted,

HICKMAN PALERMO TRUONG &
BECKER LLP

Edward A. Becker
Registration No. 37,777

1600 Willow Street
San Jose, CA 95125
(408) 414-1210
Date: July 31, 2003
Facsimile: (408) 414-1076

EXPRESS MAIL CERTIFICATE OF MAILING

"Express Mail" mailing label number EV323354645US
Date of Deposit: July 31, 2003

I hereby certify that this paper or fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to the Mail Stop Appeal Brief-Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Teresa Austin
(Typed or printed name of person mailing paper or fee)

(Signature of person mailing paper or fee)

APPENDIX

1. A method for processing data on a distributed computing system that includes a plurality of nodes, the method comprising the steps of:

maintaining mapping data that specifies work that can be performed by each of the plurality of nodes;

in response to receiving a first work request to perform first work from a first process on a first node from the plurality of nodes, determining based upon the first work and the mapping data, that the first work is to be performed on a second node from the plurality of nodes;

providing the first work request to a second process on the second node, wherein the first work request specifies that the first process is to directly receive results of the first work;

determining based upon the first work and the mapping data, that the first work is also to be performed on a third node from the plurality of nodes, and

providing a second work request to a third process on the third node, wherein the second work request specifies that results of the first work performed on the third node are to be provided directly to the first process.

2. The method as recited in Claim 1, further including the steps of

in response to receiving a second request to perform second work from the first process, determining based upon the second work and the mapping data, that the second work is to be performed on a third node from the plurality of nodes, and

providing the second request to a third process on the third node, wherein the second request specifies that the first process is to receive results of the second work directly from the third process.

3. The method as recited in Claim 1, further including the steps of

in response to receiving a second request to perform second work from a third process on a third node from the plurality of nodes, determining based upon the second work and the mapping data, that the second work is to be performed on the second node, and

providing the second request to the second process, wherein the second request specifies that the third process is to receive results of the second work directly from the second process.

4. The method as recited in Claim 1, further including the steps of

in response to receiving a second request to perform second work from a third process on a third node from the plurality of nodes, determining based upon the second work and the mapping data, a fourth node from the plurality of nodes on which the second work is to be performed, and

providing the second request to a fourth process on the fourth node, wherein the second request specifies that the third process is to receive results of the second work directly from the fourth process.

5. (CANCELED)

6. The method as recited in Claim 1, wherein the step of determining that the first work is to be performed on a second node includes the step of determining one or more resources required to perform the first work, and determining which of the plurality of nodes is allowed to perform the first work on the one or more resources.

7. The method as recited in Claim 1, wherein:
the step of determining that the first work is to be performed on a second node from the plurality of nodes includes the step of a director determining that the first work is to be performed on a second node from the plurality of nodes, and
the step of providing the first work request to a second process on the second node includes the step of the director providing the first work request to a second process on the second node.

8. The method as recited in Claim 1, further comprising the step of upon completion of the first work, the second process providing the results of the first work directly to the first process.

9. The method as recited in Claim 1, wherein the first work request is a remote procedure call.

10. (CANCELED)

11. (CANCELED)

12. (CANCELED)

13. (CANCELED)

14. (CANCELED)

15. (CANCELED)

16. (ONCE AMENDED) A distributed computing system for performing work, the distributed computing system comprising:

a plurality of nodes; and

a director communicatively coupled to the plurality of nodes, the director being configured to

maintain mapping data that specifies work that can be performed by each of the plurality of nodes,

in response to a first work request to perform first work from a first process on a first node from the plurality of nodes, determine based upon the first work and the mapping data, that the first work is to be performed on a second node from the plurality of nodes, and

request that the first work be performed by a second process on the second node, wherein the request specifies that first results of the first work be directly provided to the first process;

determining based upon the first work and the mapping data, that the first work is also to be performed on a third node from the plurality of nodes, and

providing a second work request to a third process on the third node, wherein the second work request specifies that results of the first work performed on the third node are to be provided directly from the third node to the first process.

17. The distributed computing system as recited in Claim 16, wherein the director is further configured to provide the first work request to the second process.

18. The distributed computing system as recited in Claim 16, wherein the director is further configured to

generate a second work request to request that the second process perform the first work and provide the first results directly to the first process, and

provide the second work request to the second process.

19. The distributed computing system as recited in Claim 16, further comprising resource data that specifies the access rights of the plurality of nodes relative to resources.

20. (ONCE AMENDED) A computer-readable medium carrying one or more sequences of one or more instructions for processing data on a distributed computing system that includes a plurality of nodes, the one or more sequences of

one or more instructions include instructions which, when executed by one or more processors, cause the one or more processors to perform the steps of:

maintaining mapping data that specifies work that can be performed by each of the plurality of nodes;

in response to receiving a first work request to perform first work from a first process on a first node from the plurality of nodes, determining based upon the first work and the mapping data, that the first work is to be performed on a second node from the plurality of nodes;

providing the first work request to a second process on the second node, wherein the first work request specifies that the first process is to directly receive results of the first work;

determining based upon the first work and the mapping data, that the first work is also to be performed on a third node from the plurality of nodes, and

providing a second work request to a third process on the third node, wherein the second work request specifies that results of the first work performed on the third node are to be provided directly from the third node to the first process.

21. The computer-readable medium as recited in Claim 20, further including the steps of

in response to receiving a second request to perform second work from the first process, determining based upon the second work and the mapping data, that the second work is to be performed on a third node from the plurality of nodes, and

providing the second request to a third process on the third node, wherein the second request specifies that the first process is to receive results of the second work directly from the third process.

22. The computer-readable medium as recited in Claim 20, further including the steps of

in response to receiving a second request to perform second work from a third process on a third node from the plurality of nodes, determining based upon the second work and the mapping data, that the second work is to be performed on the second node, and

providing the second request to the second process, wherein the second request specifies that the third process is to receive results of the second work directly from the second process.

23. The computer-readable medium as recited in Claim 20, further including the steps of

in response to receiving a second request to perform second work from a third process on a third node from the plurality of nodes, determining based upon the second work and the mapping data, a fourth node from the plurality of nodes on which the second work is to be performed, and

providing the second request to a fourth process on the fourth node, wherein the second request specifies that the third process is to receive results of the second work directly from the fourth process.

24. (CANCELED)

25. The computer-readable medium as recited in Claim 20, wherein the step of determining that the first work is to be performed on a second node includes the step of

determining one or more resources required to perform the first work, and

determining which of the plurality of nodes is allowed to perform the first work on the one or more resources.

26. The computer-readable medium as recited in Claim 20, wherein:

the step of determining that the first work is to be performed on a second node from the plurality of nodes includes the step of a director determining that the first work is to be performed on a second node from the plurality of nodes, and

the step of providing the first work request to a second process on the second node includes the step of the director providing the first work request to a second process on the second node.

27. The computer-readable medium as recited in Claim 20, further comprising the step of upon completion of the first work, the second process providing the results of the first work directly to the first process.

28. The computer-readable medium as recited in Claim 20, wherein the first work request is a remote procedure call.

29. (CANCELED)

30. A method for processing data on a distributed computing system that includes a plurality of nodes, the method comprising the steps of:

maintaining mapping data that specifies work that can be performed by each of the plurality of nodes; and

in response to receiving a first work request to perform first work from a first process on a first node from the plurality of nodes,

determining based upon the first work and the mapping data, that the first work is to be performed on a second node from the plurality of nodes,

generating an updated first work request that specifies that the first process is to directly receive results of performing the first work, and

providing the updated first work request to a second process on the second node.

31. An apparatus for processing data on a distributed computing system, the apparatus comprising a memory carrying one or more sequences of one or more instructions which, when executed by one or more processors, cause the one or more processors to perform the steps of:

maintaining mapping data that specifies work that can be performed by each of the plurality of nodes; and

in response to receiving a first work request to perform first work from a first process on a first node from the plurality of nodes,

determining based upon the first work and the mapping data, that the first work is to be performed on a second node from the plurality of nodes,

generating an updated first work request that specifies that the first process is to directly receive results of performing the first work, and

providing the updated first work request to a second process on the second node.

32. A computer-readable medium for processing data on a distributed computing system, the computer-readable medium carrying one or more sequences of one or more instructions which, when executed by one or more processors, cause the one or more processors to perform the steps of:

maintaining mapping data that specifies work that can be performed by each of the plurality of nodes; and

in response to receiving a first work request to perform first work from a first process on a first node from the plurality of nodes,

determining based upon the first work and the mapping data, that the first work is to be performed on a second node from the plurality of nodes,

generating an updated first work request that specifies that the first process is to directly receive results of performing the first work, and

providing the updated first work request to a second process on the second node.